

Ejemplo:

El tiempo que tarda un sistema informático en red en ejecutar una instrucción depende del número de usuarios conectados a él. Si no hay usuarios el tiempo es cero. Se tienen registrados los siguientes datos:

REALICE UN ANÁLISIS ESTADÍSTICO

- 1) Activa tu directorio de trabajo
- 2) Crea un nuevo script y llámale "Script11-DatosBivariados4"
- 3) Crea los dos vectores para las dos variables

```
# Número de usuarios = Variable explicativa o independiente
usuarios <- c(10, 15, 20, 20, 25, 30, 30)
tiempo <- c(1.0, 1.2, 2.0, 2.1, 2.2, 2.0, 1.9)
usuarios;tiempo

## [1] 10 15 20 20 25 30 30
## [1] 1.0 1.2 2.0 2.1 2.2 2.0 1.9
```

- 4) Crea una hoja de datos que tenga como componentes o columnas los dos vectores.

```
Sistema <- data.frame(Usuarios=usuarios, Tiempo=tiempo)
Sistema

##   Usuarios Tiempo
## 1      10    1.0
## 2      15    1.2
## 3      20    2.0
## 4      20    2.1
## 5      25    2.2
## 6      30    2.0
## 7      30    1.9

# Para editar o ampliar los datos puede utilizar la función fix()
fix(Sistema)
```

- 5) Guarda la hoja de datos en un archivo.

```
write.table(Sistema, file="Sistema.txt", append=FALSE, quote=TRUE, sep=" ",
            na="NA", col.names = TRUE)
```

- 6) Elimina los objetos almacenados en el área de trabajo (Workspace).

```
ls()

## [1] "Sistema" "tiempo" "usuarios"

rm(list=ls(all=TRUE))
ls()

## character(0)
```

7) Recupera la hoja de datos.

```
Sistema <- read.table("Sistema.txt", header=TRUE)
Sistema

## Usuarios Tiempo
## 1      10      1.0
## 2      15      1.2
## 3      20      2.0
## 4      20      2.1
## 5      25      2.2
## 6      30      2.0
## 7      30      1.9
```

8) Conecta la hoja de datos a la segunda ruta o lista de búsqueda.

```
attach(Sistema, pos=2)
search()

## [1] ".GlobalEnv"      "Sistema"          "package:knitr"
## [4] "package:stats"    "package:graphics" "package:grDevices"
## [7] "package:utils"    "package:datasets" "package:methods"
## [10] "Autoloads"        "package:base"
```

9) Muestra un resumen de principales estadísticos de las variables.

```
summary(Sistema)

##      Usuarios      Tiempo
## Min.   :10.00   Min.    :1.000
## 1st Qu.:17.50   1st Qu.:1.550
## Median :20.00   Median :2.000
## Mean   :21.43   Mean    :1.771
## 3rd Qu.:27.50   3rd Qu.:2.050
## Max.   :30.00   Max.    :2.200

cov(Sistema) # Matriz de covarianzas
```

```
##           Usuarios    Tiempo
## Usuarios 55.952381 2.714286
## Tiempo   2.714286 0.222381

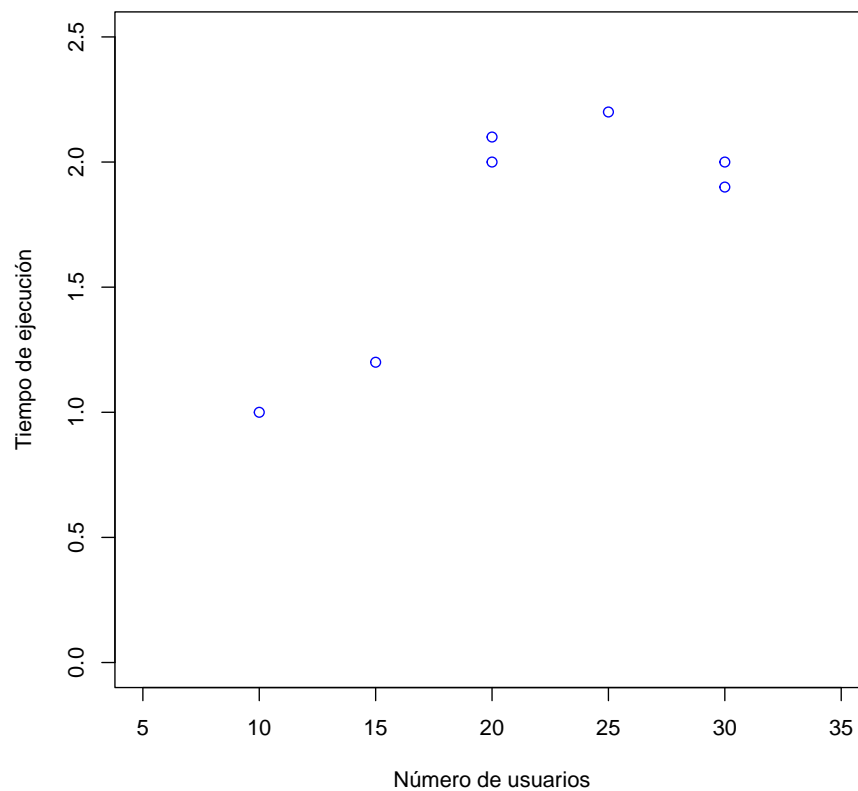
cor(Sistema, use = "all.obs", method="pearson") # Matriz de correlaciones

##           Usuarios    Tiempo
## Usuarios 1.0000000 0.7694803
## Tiempo   0.7694803 1.0000000
```

10) Elabora un gráfico de dispersión para analizar alguna relación entre las variables.

```
plot(Usuarios, Tiempo, xlim= c(5, 35), ylim= c(0.0, 2.5), type = "p", pch=1,
     col = "blue", main = "Gráfico de dispersión (Usuarios, Tiempo)",
     xlab="Número de usuarios", ylab="Tiempo de ejecución")
```

Gráfico de dispersión (Usuarios, Tiempo)



11) Para identificar un punto arbitrario, se procede de la siguiente manera:

```
#Sin cerrar la ventana del gráfico anterior, ejecuta la siguiente instrucción
identify(Usuarios, Tiempo, n=1) # n=1 indica que solamente será un punto seleccionado

## Error in identify.default(Usuarios, Tiempo, n = 1): plot.new has
not been called yet

# Y luego selecciona un punto en el gráfico haciendo clic con el ratón. Esto es útil
# para identificar puntos que podrían ser atípicos.
# Deberá aparecer en la R-Console el índice que corresponde a este punto.
```

12) Aplica la función `lm()` para encontrar el modelo lineal que se ajusta a los datos.

```
reg.Y.X <- lm(Tiempo ~ -1 + Usuarios, Sistema, na.action=NULL, method="qr", model=TRUE)
#-1 indica que no se toma en cuenta la constante en el modelo.

summary(reg.Y.X)

##
## Call:
## lm(formula = Tiempo ~ -1 + Usuarios, data = Sistema, na.action = NULL,
##     method = "qr", model = TRUE)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.4831 -0.1873  0.2056  0.3127  0.5113
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## Usuarios 0.079437    0.006496   12.23 1.82e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3871 on 6 degrees of freedom
## Multiple R-squared:  0.9614, Adjusted R-squared:  0.955
## F-statistic: 149.5 on 1 and 6 DF, p-value: 1.821e-05

# Note que es necesaria la instrucción anterior para poder visualizar los resultados más
#sobresalientes de la regresión encontrada. Nos muestra la estimación de los parámetros
#junto con su significancia, el coeficiente de determinación.
```

13) Agrega la recta de regresión al gráfico de dispersión.

```
abline (reg.Y.X)

## Error in int_abline(a = a, b = b, h = h, v = v, untf = untf, ...):
plot.new has not been called yet

#Observación: Alternativamente si quiere una recta más "exacta" use:
lines(Usuarios, 0.079437*Usuarios)

## Error in plot.xy(xy.coords(x, y), type = type, ...): plot.new has
not been called yet
```

14) Efectúa una análisis de variabilidad del modelo o descomposición de la varianza.

```
reg.anova <- anova(reg.Y.X)
reg.anova

## Analysis of Variance Table
##
## Response: Tiempo
##          Df Sum Sq Mean Sq F value    Pr(>F)
## Usuarios   1 22.4011  22.4011   149.53 1.821e-05 ***
## Residuals   6  0.8989   0.1498
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```